Chapter 14: Numerical Differentiation

Although we all remember from calculus how to analytically evaluate derivatives of a function $f(x)$, there are reasons to do this numerically. To derive formulas for numerical differentiation the basic tool is polynomial approximation. If a polynomial $p_n(x)$ approximates a given function $f(x)$ then $p_n'(x)$ approximates $f(x)$.

Chapter 14: Numerical Differentiation

We will only cover section the Taylor series approach to approximate the derivative in sections 14.1 and 14.2 this semester. There is a fundamental difficulty with numerical differentiation in that roundoff error and other high frequency noise are actually *enlarged* by the process.

14.1 Deriving formulas using Taylor series

You should be able to derive the the two-point and the three-point formulas with their orders of error using your knowledge of Taylor series. You are not responsible for the five-point formula. Write your own code to generate Table 14.1 on page 413. Do Exercise 1 in section 14.6.

14.1 Deriving formulas using Taylor series

- Two-point formula

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + \frac{h}{2} f''(\xi).$$

- Three-point formula

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{h^2}{6} f'''(\xi).$$

- Three-point formula for second derivative

$$f''(x_0) = \frac{f(x_0 + h) - 2f(x_0) - f(x_0 - h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi).$$

14.1 Deriving formulas using Taylor series

- Order one: We say, "halve the stepsize h halve the error."
- Order two: We say, "halve the stepsize h quarter the error.
- Write code to see how this works in practice.

14.2 Richardson Extrapolation

Richardson extrapolation is a simple, effective method for generating higher order numerical methods from lower ones. Given two methods of the same order, we can exploit the relationship between their leading error terms in order to eliminate such a term. We will demonstrate with an example.

14.2 Richardson Extrapolation

EXAMPLE: Use three-point approximate of f' and Richardson to get a order 4 approximate.

- three-point $f'(x_0) = F(h) + Kh^2$
- three-point $f'(x_0) = F(\frac{h}{2}) + K(\frac{h}{2})^2 = F(\frac{h}{2}) + K\frac{h^2}{4}$
- Richardson mix: $f'(x_0) = \frac{4F(\frac{h}{2}) - F(h)}{3} + Kh^4$

Chapter 15: Numerical Integration

The need to integrate a function $f(x)$ arises often in mathematical modeling. We seek approximations of the definite integral

$$\int_a^b f(x)dx \approx \sum_{j=0}^{n} a_j f(x_j)$$

for a given finite interval $[a, b]$ and integrable function f. The numerical integration formula, often referred to as a quadrature rule, has abscissae x_j and weights a_j. We will only cover sections 15.1 and 15.2 this semester.

15.1 Basic Quadrature Algorithms

In section 15.1 we derive the basic rules using polynomial interpolation

$$\int_a^b f(x)\mathrm{d}x \approx \int_a^b p_n(x)\mathrm{d}x.$$

We will discuss in class how the "Rules" and "Errors" in the green box on page 445 are derived from the Lagrange polynomials in Chapter 10.

15.1 Basic Quadrature Algorithms

You should understand how the "Rules" and "Errors" are used in Example 15.2 on page 446. You should be able to write Python code to generate the same values from this example.

15.1 Basic Quadrature Algorithms

- Midpoint Rule: $(b-a)f(\frac{a+b}{2})$, Error: $\frac{f''(\xi)}{24}(b-a)^3$.
- Trapezoid Rule: $\frac{b-a}{2}(f(a)+f(b))$, Error: $-\frac{f''(\xi)}{12}(b-a)^3$.
- Simpson's Rule: $\frac{b-a}{6}(f(a)+4f(\frac{a+b}{2})+f(b))$, Error: $\frac{f^{(4)}(\xi)}{90}(b-a)^5$.

15.2 Composite Quadrature Algorithms

- Composite Trapezoid Rule:

$$\int_a^b f(x)dx \approx \frac{h}{2}\left(f(a) + 2f(t_1) + 2f(t_2) + \ldots + f(t_{r-1} + f(b)\right)$$

  Error: $-\frac{f''(\nu)}{12}(b-a)h^2$.

- Simpson's Rule:

$$\int_a^b f(x)dx \approx \frac{h}{3}\left(f(a) + 4f(t_1) + 2f(t_2) + 4f(t_3) + \ldots + f(b)\right)$$

  Error: $-\frac{f^{(4)}(\psi)}{180}(b-a)h^4$.

15.2 Composite Numerical Integration

You only need to know how the Composite Trapezoid, Composite Midpoint, and Composite Simpson's rule are generated from the basic algorithms in section 15.1. You should know how to use the "Quadrature Errors Theorem" on page 453 but you need not know how to derive these formulas. These formulas are good to have on your formula sheet for the final exam. You should work out Exercises $1, 3, 5(a)$ from section 15.7. You are not responsible for sections $15.3 - -15.6$.

Chapter 16 Differential Equations

This is review of our work at the beginning of the semester studying the SIR initial value problem (IVP or Initial Value Problem). The SIR model of infectious disease is a nonlinear system of differential equations. It is unsolvalbe in that we are unable to write formulas $S(t), I(t)$, and $R(t)$ in terms of the basic functions $y = t^2, y = \sin t, y = \log t, \ldots$ we learned in calculus. Yet the existence and uniqueness theorem of differential equations proves that there are functions $S(t), I(t)$, and $R(t)$.

Chapter 16 Differential Equations

Since we are unable to write formulas for $S(t), I(t)$, and $R(t)$, it does not mean that we cannot solve them. We can approximate these solutions. Solving such problems is one of the main reasons you are taking math 328. The algorithms used to solve such IVPs are ubiquitous in scientific computing. At the beginning of the semester we used Euler's method to approximate $S(t), I(t), R(t)$. In this chapter we will see that there are better, i.e. higher order, methods to approximate solutions to IVPs. We will study two such methods: RK2 and RK4.

Chapter 16 Differential Equations

You should only use Python to generate table 16.2 on page 496. and Figure 16.5 on page 497. You should know that Euler's method is an order 1 method, RK2 is an order 2 method, and RK4 is an order 4 method. If you have not already done so please watch the Lecture 2 video from Arthur Mattuck's MIT Differential Equations course (there is a link to this video on our coursepage). You need not read this chapter or do any exercises. Be organized and systematic when doing Euler's, RK2, or RK4 by hand. As always make a couple iterations by hand before employing a computer.

Section 16.1, 16.2, and 16.3

- Euler's Method is an order one method. If we cut the stepsize in half then we will "roughly" cut the error in half.
- Trapezoid Rule (RK2) is an order two method. If we cut the stepsize in half we will "roughly" cut the error by $\frac{1}{4}$.
- RK4 is obviously of 4th order. If we cut the stepsize h in half using RK4 then the error will "roughly" be cut in $\frac{1}{16}$th.

Chapter 16 Differential Equations

You should memorize the orders of the three methods on the previous slide. These are the only methods you will need for the final exam. You have already used Euler's method extensively in the SIR model. The other two methods are variants of Euler's method. It is best to think of Euler's method as the blueprint for the other two higher order methods as explained in Arthur Mattuck's video from MIT on our coursepage.